492-MOC-004

DRAFT

GAMMA-RAY LARGE AREA SPACE TELESCOPE (GLAST) PROJECT

Procedure/PROC Style Guide

Version 1.00

July 16, 2004

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

CHECK THE GLAST PROJECT WEBSITE AT
http://glast.gsfc.nasa.gov/project/cm/mcdl TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

492-MOC-004

DRAFT

GAMMA-RAY LARGE AREA SPACE TELESCOPE
(GLAST)
PROJECT

Procedure/PROC Style Guide

July 17, 2004

NASA Goddard Space Flight Center

Greenbelt, Maryland

CHECK THE GLAST PROJECT WEBSITE AT
http://glast.gsfc.nasa.gov/project/cm/mcdl TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DRAFT

GLAST PROJECT PROCEDURE/PROC STYLE GUIDE

Prepared by:

Jonathan DeGumbia MOC System Engineer	Date

Concurrence:

Robin Corbet GSSC Operations Lead	Date
David Lung LAT IOC Subsystem Manager	Date
William Paciesas GBM Ground System Lead	Date
Kenneth Lewis GLAST Operations Lead (acting)	Date
Doug Spiegel GLAST MOC Manager	Date

Approved by:

Ken Lehtonen GLAST Ground System/Ops Manager	Date
Dennis Small GLAST MOC Lead	Date

DRAFT

REVISION STATUS

VERSION	DATE	CHANGED BY	DESCRIPTION
0.01	6/2/2004	J. DeGumbia	Preliminary document
0.02	7/6/2004	J. DeGumbia	Internal Review version
0.03	7/9/2004	J. DeGumbia	Baseline Review version
1.00	7/16/2004	J. DeGumbia	Baseline Submission

TABLE OF CONTENTS

1 Introduction 1
1.1. Purpose 1
1.2. Scope 1
1.3. Document Organization 1
1.4. Applicable Reference Documents 1
1.4.1. Referenced Documents 1
1.4.2. Applicable Documents 1
2 Overview 2
2.1. Philosophy of Operations 2
2.2. Development Flow Overview 2
3 Style Guides 3
3.1. Procedures 3
3.1.1. File Format 3
3.1.2. File Naming Convention 3
3.1.3. Procedure Format 3
3.1.3.1. Procedure Header (Mandatory) 3
3.1.3.2. Procedure Footer (Mandatory) 3
3.1.3.3. Revision Summary Table (Mandatory) 4
3.1.3.4. Purpose (Mandatory) 4
3.1.3.5. Description (Mandatory) 4
3.1.3.6. Reference (Mandatory) 4
3.1.3.7. Initial Configuration / Entrance Criteria (Mandatory) 4
3.1.3.8. Final Configuration / Exit Criteria (Mandatory) 4
3.1.3.9. Constraints (Mandatory) 5
3.1.3.10. Parent Procedure (Mandatory) 5
3.1.3.11. Child Procedure(s) (Mandatory) 5
3.1.3.12. Relevant Modes of Operation (Optional) 5
3.1.3.13. Estimated Duration (Optional) 5
3.1.3.14. Procedure Script 5
3.1.3.14.1. Table Columns 5
3.1.3.14.2. Table Sections 6
3.2. PROCs 7
3.2.1. File Format 7
3.2.2. File Naming Convention 7
3.2.3. Format 7
3.2.3.1. Name 7
3.2.3.2. Header 7
3.2.3.3. Body 9
Appendix A: Procedure Example 10
Appendix B: PROC Examples 16

Acronyms

ASCII	American Standard Code for Information Interchange
CVS	Concurrent Versions System
FOT	Flight Operations Team
GBM	Gamma-ray Burst Monitor
GLAST	Gamma ray Large Area Space Telescope
ITOS	Integrated Test and Operations System
I&T	Integration and Test
LAT	Large Area Telescope
MOC	Mission Operations Center
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
NSSTC	National Space Science & Technology Center
PDB	Project Database
SAI	Spectrum Astro, Incorporated
SLAC	Stanford Linear Accelerator Center
STOL	Systems Test and Operations Language
T&C	Telemetry and Command

1 Introduction

1.1. Purpose

The purpose of the *GLAST Project Procedure/PROC Style Guide* is to provide a series of guidelines that will ultimately enable various disassociated parties to implement GLAST operational PROCS with efficiency and commonality.

1.2. Scope

The *GLAST Project Procedure/PROC Style Guide* is the single repository for information concerning the format by which Procedures and PROCs will be developed. All parties involved with these duties are expected to adhere to the conventions contained within.

1.3. Document Organization

This document is logically divided on 5 main sections:

- Section 1 provides necessary, ancillary information to aid the reader in understanding this document and its place in the GLAST program.
- Section 2 provides fundamental information about GLAST Procedures and PROCS.
- Section 3 provides a detailed set of conventions that will guide the reader in the development of the *Procedures* and *PROC*s.
- Appendix A provides a sample of a *Procedure* that follows the guidelines contained within this document.
- Appendix B provides samples of *PROC*s that follow the guidelines contained within this document.

1.4. Applicable Reference Documents

1.4.1. Referenced Documents

The following documents are referenced in the Mission Operations Center (MOC) requirements:

- GLAST On-orbit Operations Description Manual
- ITOS Users Guide, On-line Documentation, http://itos.gsfc.nasa.gov/

1.4.2. Applicable Documents

The following documents are listed for reference.

- GLAST Spacecraft-MOC Interface Control Document
- 492-MOC-006: GLAST Configuration Management Plan

2 Overview

2.1. Philosophy of Operations

In effort to control the inherent risk involved with sending potentially harmful commands to the observatory during the Integration and Test (I&T) and operational phases, commands will be sent to the observatory by executing PROCs. PROCs are electronic files that contain ITOS STOL code consisting of logic and control statements, telemetry checks, comments, and commands. PROCs provide a means for safe and consistent observatory commanding by allowing for prerequisite checks of the observatory configuration, post-command telemetry checks of expected observatory configuration, and configuration controlled repeatability.

2.2. Development Flow Overview

The Goldbelt Orca/Omitron FOT is responsible for writing all observatory PROCs. The FOT will write the PROCs according to the specifications supplied by each of the three organizations who maintain the necessary expertise. These organizations are the Stanford Linear Accelerator Center (SLAC) for the LAT related PROCs, the National Space Science & Technology Center (NSSTC) for GBM related PROCs, and Spectrum Astro, Incorporated (SAI) for spacecraft bus related PROCs. Section 3.2 contains a detailed description of PROCs. An example of a PROC may be found in Appendix B.

To facilitate communications between the various organizations and the FOT, the PROC specifications will be defined in documents known as Procedures. A Procedure is an electronic file that contains all of the steps necessary to accomplish a given task on the observatory. The Procedures are not expected to have a one-to-one relationship with the PROCs. Section 3.1 contains a detailed description of Procedures. An example of a Procedure may be found in Appendix A.

The three contributing organizations are responsible for creating all of the Procedures necessary to operate the observatory during on-orbit operations phases. SAI will present all Procedures to the GLAST project as part of the *On-orbit Operations Description Manual*. The FOT will then use these Procedures to create operational PROCs.

3 Style Guides

3.1. Procedures

3.1.1. File Format

Each Procedure will be saved to a separate file. Procedures are to be saved in Microsoft Word file format.

3.1.2. File Naming Convention

The filename for each Procedure should contain useful information that will help to identify the function it performs. Filenames should be limited to 16 characters. All characters are allowed except for (\ / : * ? " < > |). Procedure filenames should have a *.doc* file extension.

The first four letters of each Procedure will provide indication of its origin.

- Procedures dealing primarily with the spacecraft bus will begin with "BUS_"
- Procedures dealing primarily with the LAT will begin with "LAT_"
- Procedures dealing primarily with the GBM will begin wit "GBM_"

The characters following the first four should be selected so as to provide a clear indication of its purpose.

3.1.3. Procedure Format

The following sections describe the various informational fields that are to be included in the Procedures. A sample Procedure may be found in Appendix A of this document.

3.1.3.1. Procedure Header (Mandatory)

The Procedure header will appear on every page of the Procedure. It contains the following information:

Procedure Number – The Procedure Number appears left-justified in the header. Its format is not restricted. Each Procedure must have a unique Procedure Number.

Procedure Name – The Procedure Name appears centered in the header. Its format is not restricted. The Procedure name should be similar to the Procedure file name, and it should give a clear indication of the function the Procedure performs.

Version Number and Date – The Version Number and Date appear right-justified in the header. This information is used to identify one Procedure version from another.

3.1.3.2. Procedure Footer (Mandatory)

The Procedure footer will appear on every page of the Procedure. It contains the following information:

Mission Name – The Mission Name appears left-justified in the footer. The name *GLAST* will be used.
Filename – The Filename appears centered in the footer. The filename will be the name of the file, including the file extension (.doc), which contains the Procedure details.

Page Number – The Page Number appears right-justified in the footer. The format of the page number will be: Page X of Y. Page numbers will begin with 1 and count in whole number increments.

3.1.3.3. Revision Summary Table (Mandatory)

The Revision Summary Table will track information regarding the change history of the Procedure. The Table will consist of three columns:

Rev – The Rev column will contain the version number associated with line entry.

Date – The Date column will contain the date that the change was made.

Description of Change – The Description of Change will contain a concise overview of the modification made to the Procedure.

A new row will be added to the table each time a change is made to the Procedure.

3.1.3.4. Purpose (Mandatory)

The Purpose section contains a detailed description of the reason the Procedure is needed.

3.1.3.5. Description (Mandatory)

The Description section contains a detailed description of the function(s) the Procedure will perform. Multiple functions will be listed numerically in the order of execution (where possible).

3.1.3.6. Reference (Mandatory)

The Reference section contains the names of any documents where relevant information about the function(s) being performed may be found.

3.1.3.7. Initial Configuration / Entrance Criteria (Mandatory)

The Initial Configuration / Entrance Criteria section contains a numeric list of observatory configurations that the spacecraft must be in before the Procedure may begin. The configuration checks should be made to be human readable. Pseudo-code may be used if the author feels the reader will benefit from it. Mnemonics should be avoided.

3.1.3.8. Final Configuration / Exit Criteria (Mandatory)

The Final Configuration / Exit Criteria section contains a numeric list of observatory configurations that the spacecraft is expected to be in following the successful execution of the Procedure. The configuration checks should be made to be human readable. Pseudo-code may be used if the author feels the reader will benefit from it. Mnemonics should be avoided.

3.1.3.9. Constraints (Mandatory)

The Constraints section contains a numeric list of restrictions or limitations that apply during the execution of the Procedure. It is in this section that any additional information relating to the safe execution of the Procedure is captured. The constraints should be made to be human readable. Pseudo-code may be used if the author feels the reader will benefit from it. Mnemonics should be avoided.

3.1.3.10. Parent Procedure (Mandatory)

The Parent Procedure contains the Procedure Number and Procedure Name of the calling Procedure. If no parent Procedure exists, enter *N/A*.

3.1.3.11. Child Procedure(s) (Mandatory)

The Child Procedures section contains a numerical list of any Procedures called by the Procedure. If no child Procedures exist, enter *N/A*.

3.1.3.12. Relevant Modes of Operation (Optional)

The optional Relevant Modes of Operation section contains a numeric list of the different modes under which the Procedure may be executed.

3.1.3.13. Estimated Duration (Optional)

The optional Estimated Duration section contains the expected duration of time required to execute the Procedure. Any assumptions or caveats associated with the estimate should be included.

3.1.3.14. Procedure Script

The Procedure Script is a stepwise instructional Procedure detailing the operations that the on-line personnel must perform to execute the Procedure. The script is presented in the form of a table, where the rows indicate the steps and the columns indicate details about each step. The organization of the table is detailed in the following subordinate sections.

3.1.3.14.1. Table Columns

STEP – The Step column provides the order and a point of reference for the rows that indicate action (i.e. not header rows). Each step within a table section should have unique number. Numbers within each table section should begin with *1* and increase in whole number increments.

DESCRIPTION – The Description column indicates what the step is supposed to do.

<u>ACTION</u> – The Action column indicates precisely what the on-line personnel should do. It could include queuing and sending command mnemonics, opening telemetry pages, selecting menu items, etc.

<u>EXPECTED RESPONSE</u> – The Expected Response column provides positive indications of a successfully completed action. It may include expected telemetry mnemonic values/states or software responses.

<u>COMMENTS</u> – The Comment column may be used at the discretion of the author to add any additional information concerning the step. This information may include relevant contingency Procedures, reference material, or a justification for the expected response.

3.1.3.14.2. Table Sections

For purposes of readability the table is divided into various sections. Each section is prescribed a letter and a name. Each section within the table must have a unique letter designation and name. Section letters begin with *A* and continue alphabetically. Table sections are delineated by header rows. Header rows span the entire width of the table and contain the prescribed letter and name.

Of the sections, only the first and last are defined. The first section should be labeled *Part A. – Initial Configuration*. The purpose of the Initial Configuration section is to document the expected state of the observatory and/or ground support equipment prior to executing any changes. While executing the Procedure, on-line personnel will perform the configuration verification during these initial steps. Configuration verifications should be limited to checking states that have direct impact on the success of the Procedure.

The last section should be labeled *Part* x. *– Final Configuration*. The Final Configuration section documents the expected state of the observatory and/or ground support equipment following the execution of all changes within the Procedure. While executing the Procedure, on-line personnel will perform the configuration verification during these final steps. Configuration verifications should be limited to checking states that ensure the Procedure was successfully executed and observatory and/or ground support equipment has responded as expected

The section(s) between the Initial Configuration and Final Configuration are left to the Procedure author to define. These sections should contain all of the observatory and/or ground support equipment configuration changes within the Procedure.

3.2. PROCs

3.2.1. File Format

Each PROC will be saved to a separate file. PROC are to be saved in ASCII text file format.

3.2.2. File Naming Convention

The filename for each Procedure should contain useful information that will help to identify the function it performs. Filenames should be limited to 16 characters. All characters are allowed except for (\ / : * ? " < > |). Procedure filenames should have a *.txt* file extension.

3.2.3. Format

PROC files will be executed, without conversion, on the ITOS T&C system. As such, each PROC will consist solely of STOL code. Information concerning the details of STOL logic statements may be found in the *ITOS Users Guide*.

Within the confinements of the ITOS STOL language, PROCs used for the GLAST program will exhibit additional refinements. The following conventions should apply to all text within a PROC:

- ITOS directives will appear in upper case.
- No tabs will be used.
- Blank lines will be allowed but a comment ";" is encouraged instead for readability.
- Strings beginning and ending with dollar signs such as *xyz*, are reserved for CVS keywords and should be strictly avoided.

In support of consistency and completeness, PROCs within the GLAST program will also adhere to a specified format. The following sections describe the various informational fields and associated constraints that are to be included in the GLAST PROCs. A sample PROC may be found in Appendix B of this document.

3.2.3.1. Name

The first line of a PROC will appear as: `PROC [Procedure_Name]`, where the actual name of the PROC, omitting the *.txt* file extension will be substituted for the bracketed phrase. The PROC name should be selected so as to provide some indication as to its purpose.

3.2.3.2. Header

The header will immediately follow the PROC name. It consists of a series of comment statements that contain information about the PROC file itself. See Appendix B for details concerning the Header layout. Minor style differences throughout this and other

PROC fields are expected and will be tolerated. The following information must be contained within the Header.

Mission Name – The mission name line will appear as:

```
;   MISSION: GLAST.
```

Title – The title line provides the title of the PROC. It may be different and more descriptive than the PROC file name.

Author – The author line will show the name of the creator of the original version of the PROC.

Date – The date line will provide the month, day, and year that the original version of the PROC was completed. The date will appear in *month dd, yyyy* format.

Purpose – The purpose will contain a concise description of the reason the PROC is needed and function(s) the PROC performs. If multiple functions are performed, they will be listed in the order of execution (where possible).

Responsible Lead – The responsible lead line will indicate the name and position of the individual who possesses responsibility for PROC.

PROC Call Outs: The PROC call outs field will contain a table of all PROCs that can be initiated during the execution of the PROC. The table will consist of a Name column that specifies the child PROC by name and an Activity column that briefly states the function that is accomplished by that PROC.

Revision History – The revision history field will provide an overview of the evolution of the PROC. It consists of two areas: The current version line and the revision history line.

The current version line should appear as:

```
;   $Id$ $Name$
```

When the PROC is released from the CVS repository, information such as the PROC filename, version number, and check-in date will appear in the PROC.

Similarly, the revision history line should appear as:

```
;   $Log$
```

CVS will replace this line with a revision history table that will contain a chronological list of changes made to the PROC.

<u>Associated Procedure</u>: The related Procedure line will contain the name of the delivered Procedure, if any, used create the PROC. If no related Procedure exists, add the word `NONE` in place of the Procedure Name.

<u>Execution Time</u>: The execution time line contains the best available estimate of the time required to execute the PROC under normal operational conditions.

3.2.3.3. Body

The PROC body contains the series of STOL statements and comments necessary to achieve the purpose of the PROC as stated in the header. The following conventions should be respected:

- The following line should appear at the beginning of the body of the PROC:
  ```
  SHO "revision: $Id$ $Name$"
  ```
- Comments will be used liberally to promote organization and readability
- Local and global variables will appear in lower case.
- Command, telemetry, or ground system mnemonics or sub mnemonics will appear in uppercase.
- Labels will appear in upper case. All labels will be called in uppercase.
- The indentation for all loops and IF statements will be four spaces.
- Statements in ASK and WRITE statements as well as comments will appear in upper and lower case as necessary to promote readability.
- Nesting will be limited to five levels.
- Structured programming is encouraged and GOTOs will be limited.
- Commanding will be in "ONESTEP" mode.
- Relational operators like ".EQ.", ".AND.", and ".OR." will be in uppercase and have spaces around them to improve readability as in the following example:
  ```
  WAIT UNTIL (MNEMONIC1 .EQ. (MOD((localex + 1),256)))
  ```
- PROCs destined for operations reuse will contain command end item telemetry verification using "WAIT UNTIL" statements.
- Parameters passed to the PROC will be validated whenever possible.
- ASK statements requiring discrete input or known ranges will be structured in a DO WHILE loop that will check for correct responses.
- PROCs will end with the following two lines:
  ```
  SHO "PROC [PROCNAME] COMPLETED."
  ENDPROC
  ```

Appendix A: Procedure Example

REVISION SUMMARY		
Rev	Date	Description of Change
1.0	04/01/1999	Initial release.
1.1	04/25/1999	Incorporated timing modifications from Rehearsal #1
5.2	03/02/2000	Change in version control methodology: - Version indicates the release of L&EO Plan - Date indicates date of last modification to actual procedure within the LEO Plan
5.2	06/01/2000	Removed dependence on lower ACS mode. Procedure may now be executed in any ACS mode equivalent to or higher than Sun Point.

Purpose:

1. Procedure places the SC in LVLH (Local Vertical Local Horizontal) ACS Mode and check for tracking of the Solar Array actuators (Solar Array Drive Assembly or SADA).

Description:

Procedure performs three functions (one command function, two telemetry verification functions):

1. Transitions SC to LVLH from any ACS Mode $\geq$ Sun Point (Mode $\geq$ 3)
2. Verifies proper function of ACS (FSW and HW)
3. Verifies proper function of Solar Array Drive Assembly

Reference:

1. On-Orbit Handbook, section 5.1.4 ACS Mode Transitions
2. Space Vehicle Handbook, section 2.5.3.1 SADA Diagnostic Telemetry

Initial Configuration / Entrance Criteria:

1. Nominal execution of current ACS mode
   - This procedure is not to be executed as a contingency procedure. In the event of a non-optimal ACS performance reference ACS Contingency Flow Diagram (SV-ACS-C001-FlwDgm).
2. Initial ACS Mode $\geq$ Sun Point (Mode $\geq$ 3)
3. If entering LVLH from Sun Point then:
   a. Updated ephemeris on-board and propagating for a minimum of 20 minutes prior to ACS transition to LVLH.
   b. Updated ephemeris on-board and propagating no more than 12 hours at the time of commanded ACS transition to LVLH.
4. SC in sunlight at the time of commanded ACS transition to LVLH and SC in sunlight for a minimum of 10-minutes following commanded ACS transition to LVLH.
5. Nominal EPS performance
   - This procedure is not to be executed as a contingency procedure. In the event of a non-optimal EPS performance reference EPS Contingency Flow Diagram (SV-EPS-C001-FlwDgm).

6. Minimum of three minutes remaining in support prior to scheduled LOS at the time of commanded ACS transition to LVLH.
7. Safe Mode Detection Sensor Enabled (ASMDETSNSR = ENBLD)

Final Configuration / Exit Criteria:

1. SC in ACS Mode = LVLH
2. Solar Arrays rotating at nominal rate tracking the Sun
3. EPS Solar Array Currents nominal.

Constraints:

1. Initial ACS Mode ≥ Sun Point
    - If ACS cannot achieve and maintain Sun Point then there is a serious SC anomaly. The potential ramifications of any transitions to an ACS mission mode (Mode > 3) should be carefully considered and is out of the scope of this procedure.

2. On-board Propagator initialized and propagating with valid ephemeris prior to ACS transition to LVLH
    - ACS FSW propagates a SC ephemeris as long as the CPU is on and ACS FSW is operating nominally. The ACS FSW Attitude Control algorithm only utilizes this propagated vector as an input when operating at an ACS Mission Mode (Mode > 3). If the on-board ephemeris is invalid in a mission mode then, over time, the vehicle will detect a diverging attitude solution based on sensor inputs compared to the propagated ephemeris and trip a Safe Mode entry condition. This is non-optimal and is not recommended by Spectrum Astro, Inc.
3. SC in Sunlight at the time of commanded ACS transition
    - If a Safe Mode entry condition is tripped upon transition to LVLH then the SC will fall to ACS Inertial Acquisition Mode (Mode = 1) and perform a set of Sun Search pitch and roll maneuvers. If the Sun is not present, then the SC will continue to perform these maneuvers until a Coarse Sun Sensor (CSS) detects Sun Presence at which time it will transition to ACS Sun Acquisition (Mode = 2). This is non-optimal and is, as a general practice, not recommended by Spectrum Astro, Inc. Any intended execution of this procedure during Eclipse from the Sun should be pre-coordinated with Spectrum Astro, Inc. Mission Operations personnel.
4. Minimum remaining contact at time of commanded ACS transition
    - In general, it is a good practice that this activity be performed with view of the ground operations personnel. It is also a good practice that the ground personnel monitoring performance of the activity leave enough time to respond to an anomalous occurrence during the SC mode transition activity. It is therefore not recommended by Spectrum Astro, Inc. to perform the mode transition if there is less than 3 minutes remaining in the support because it does not allow what we deem to be appropriate for responding to an anomalous occurrence upon attempted ACS mode transition.
5. Safe Mode Detection Sensor Enabled

- If ACS is unable to maintain a nominal LVLH mode execution, then without the Safe Mode Detection Sensor Enabled, the ACS FSW will not have a mechanism to Safe itself. The SC has other mechanisms in the event of a power under-voltage (UV Trip Levels, reference OOH section 10.1), however, it is considered a bad practice to disable and leave disabled higher-level safing mechanisms that are functioning properly. It is highly recommended that the Safe Mode Detection Sensor be enabled prior to execution of this activity. Any plans to perform this activity without the Safe Mode Detection Sensor enabled should be pre-coordinated with Spectrum Astro, Inc.

2. Anomalous occurrences may be identified anywhere from 10 seconds to 90-minutes after the start of the procedure.

Parent Procedure:

1. SV-LEO-002 (Launch & Early Orbit Mode, Post-Initialization Activity Flow Diagram)

Child Procedure(s):

1. SV-ACS-C001 (ACS Contingency Flow Diagram)
2. SV-EPS-C001 (EPS Contingency Flow Diagram)

Relevant Modes of Operation:

1. L&EO Mode
2. Safing Mode
    a. Post-Diagnosis Recovery Operations

Estimated Duration:

1. Nominally procedure may be completed in 6 minutes 30 seconds

STEP	DESCRIPTION	ACTION	EXPECTED RESPONSE	COMMENT
Part A. – Initial Configuration				
1.	Initial Configuration/Entry Criteria as defined by procure header excluding valid ephemeris.		MODE ≥ 3 SUNPRESENCE = YES EBUSV ≥ 28.0 V ACSSCNT1 ≥ 125 ASMDESNSR = ENBLD	If Expected Response is violated, then ABORT procedure due to invalid entry to ACS mode transition.
2.	Verify on-board ephemeris is propagating nominally relative to anticipated position (X,Y,Z ECI)		APROP = PROPAGATION AEPHMPOSX = Nominal XPOS +/- 10000 m AEPHMPOSY = Nominal YPOS +/- 10000 m AEPHMPOSZ = Nominal YPOS +/- 10000 m	If on-board ephemeris is outside of limits then an updated State Vector Upload is required prior to continuation to step B1 (refer to ACS-005).
Part B. – Command ACS Mode Transition				
1.	Load SW command to transition to ACS Mode = LVLH	>> ***MODE = 5 (*** OFFSET_ROLL = 0.0 OFFSET_PITCH = 0.0 OFFSET_YAW = 0.0)	INQCHK = 0x22020005	

STEP	DESCRIPTION	ACTION	EXPECTED RESPONSE	COMMENT
2.	Following checksum verification, execute input queue. Command sequence transitions ACS to LVLH/Mission Mode	>> *EXINQ_RT*	MODE =5 (LVLH/MISSION) MODELIM = 5 (LVLH/MISSION)	Mnemonic MODELIM should change to LVLH/MISSION instantaneously. MODE change to LVLH/MISSION requires the ACS to meet the set of criteria as defined in ACS SVH. If Expected Response does not occur within 2 minutes, then CALL contingency procedure: **SV-ACS-C001**
Part C. – Verify Solar Arrays Tracking the Sun				
1.	Following ACS Mode transition (MODE = LVLH/MISSION), allow approximately 20 seconds to verify that the Solar Arrays are tracking the Sun.	Wait up to 20 seconds	SADAB1STP > TBD and incrementing SADAB2STP > TBD and incrementing SADAB1DEG > TBD and incrementing SADAB1DEG > TBD and incrementing	SADAs should rotate to track the sun within 20 seconds of transition to LVLH. If Expected Response does not occur within 20 seconds, then CALL contingency procedure: **SV-EPS-C001**

STEP	DESCRIPTION	ACTION	EXPECTED RESPONSE	COMMENT
Part D. – Final Configuration				
1.	Verify Final Configuration/Exit Criteria.	Wait up to 20 seconds	Mode = 5 SADAB1STP incremtenting at ~ 0.1 counts per second ESAI > 0.1 or EBATI > 0.0	Assuming 495 km circular orbit, in LVLH Solar Arrays should rotate at 4-8 counts per minute. If solar arrays are generating power properly and vehicle is in sun (Entry Criteria) then SA current should be greater than 0.1A unless the Battery is in trickle charge. If Mode or SADAB1STP is not equivalent to Expected Response, then CALL contingency procedure: **SV-ACS-C001** If EPS is not equivalent to Expected Response, then CALL contingency procedure: **SV-EPS-C001**

Appendix B: PROC Examples

```
PROC PROC_HEADER
;--------------------------------------------------------------------------
;
;   MISSION: GLAST
;
;   TITLE:
;
;   AUTHOR:
;
;   DATE:
;
;   PURPOSE:
;
;--------------------------------------------------------------------------
;
;   RESPONSIBLE LEAD:
;
;--------------------------------------------------------------------------
;
;   PROC CALL OUTS:
;
;       NAME:              ACTIVITY:
;     ------------        ----------------------
;
;
;
;--------------------------------------------------------------------------
;
;   REVISION HISTORY:
;
;    $Id$ $Name$
;
;    $Log$
;
;--------------------------------------------------------------------------
;
;   ASSOCIATED PROCEDURE:
;
;--------------------------------------------------------------------------
;
;   EXECUTION TIME:
;
;**************************************************************************
;
;---------------------- DEFINE PROCEDURE VARIABLES -----------------------
;
SHO "Revision: $Id$ $Name$"
;
GLOBAL
LOCAL
;
;
SHO "PROCEDURE PROC_HEADER COMPLETE."
ENDPROC
```

```
PROC NIFIXOPEN
;--------------------------------------------------------------------------
;
;   MISSION: GLAST
;
;   TITLE:   NISTAR Set Fixed Open Precharge
;
;   AUTHOR:  Joseph Kowalski, Honeywell Technology Solutions Inc.
;
;   DATE:    April 21, 2000
;
;   PURPOSE: This procedure sets the fixed open precharge for the
;            drop floor algorithm. It should only be run when NISTAR
;            is in Science Mode.
;
;--------------------------------------------------------------------------
;
;   RESPONSIBLE LEAD:  Dr. Steven Lorentz, NISTAR PI
;
;--------------------------------------------------------------------------
;
;   PROC CALL OUTS:
;
;       NAME:       ACTIVITY:
;     ------------    ---------------------
;
;
;--------------------------------------------------------------------------
;
;   REVISION HISTORY:
;
;    $Id$ $Name$
;
;    $Log$
;
;--------------------------------------------------------------------------
;
;   ASSOCIATED PROCEDURE:
;
;--------------------------------------------------------------------------
;
;   EXECUTION TIME:
;
;**************************************************************************
;
;---------------------- DEFINE PROCEDURE VARIABLES ------------------------
;
SHO "Revision: $Id$ $Name$"
;
LOCAL cmdcnt, cmderrcnt, device, filter, setpt
;
IF (P@NIINSTMODE .NE. "SCIENCE") THEN
    SHO "Instrument not in Science mode"
    GOTO END
ENDIF
;
ASK "Hit 'OK' to configure fixed open precharge"
;
cmdcnt = NICMDCNT ; Command Accept Counter
cmderrcnt = NIERRCNT ; Command Reject Counter
;
device = 1 ; Initalize variable
filter = 1 ; Initalize variable
setpt = "x" ; Initalize variable
;
DO UNTIL ((device .EQ. "RC1") .OR. (device .EQ. "RC2") .OR. (device .EQ. "RC3"))
    ASK "Select the device (RC1/RC2/RC3)", device
    device = UPPERCASE(device)
ENDDO
;
```

```
DO UNTIL ((filter .EQ. "A") .OR. (filter .EQ. "B") .OR. (filter .EQ. "C"))
    ASK "Select the filter (A/B/C)", filter
    filter = UPPERCASE(filter)
ENDDO
;
ASK "Enter the setpoint for the precharge", setpt
;
/NIFOPRECHRG DEVICE=name(device), FILTER=name(filter), SETPOINT=(setpt)
;
WAIT UNTIL (NICMDCNT .EQ. MOD((cmdcnt + 1),256) .AND. (NIERRCNT .EQ. cmderrcnt))
IF ((device .EQ. "RC1") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC1FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC1") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC1FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC1") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC1FOPRECHRGC .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC2FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC2FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC2") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC2FOPRECHRGC .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "A")) THEN
    WAIT UNTIL (NIRC3FOPRECHRGA .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "B")) THEN
    WAIT UNTIL (NIRC3FOPRECHRGB .EQ. setpt)
ELSEIF ((device .EQ. "RC3") .AND. (filter .EQ. "C")) THEN
    WAIT UNTIL (NIRC3FOPRECHRGC .EQ. setpt)
ENDIF
;
END:
SHO "PROCEDURE NIFIXOPEN COMPLETED."
ENDPROC
```

```
PROC TCATSSTART
;---------------------------------------------------------------------------
;
;   MISSION: GLAST
;
;   TITLE:   Start ATS
;
;   AUTHOR:  Test Conductor
;
;   DATE:    10/20/2000
;
;   PURPOSE: Start ATS buffer
;
;---------------------------------------------------------------------------
;
;   RESPONSIBLE LEAD:  C&DH Lead
;
;---------------------------------------------------------------------------
;
;   PROC CALL OUTS:
;
;       NAME:                ACTIVITY:
;     ------------        ----------------------
;
;
;---------------------------------------------------------------------------
;
;   REVISION HISTORY:
;
;    $Id$ $Name$
;
;    $Log$
;
;---------------------------------------------------------------------------
;
;   ASSOCIATED PROCEDURE:
;
;---------------------------------------------------------------------------
;
;   EXECUTION TIME:
;
;***************************************************************************
;
;------------------------ DEFINE PROCEDURE VARIABLES ------------------------
;
SHO "Revision: $Id$ $Name$"
;
LOCAL  ans, answer, cmdctr
;
;  START THE ATS
;
; SELECT WHICH ATS BUFFER TO START, A OR B
;
ans="x"
DO WHILE ((ans .NE. "A") .AND. (ans .NE. "B") .AND. (ans .NE. "N"))
    ASK "Start ATS in buffer A OR B (A/B) OR N = NO to starting a buffer", ans
    ans = UPPERCASE(ans)
ENDDO
;
IF ((ans .EQ. "A") .AND. (SCATPSTATE .NE. 2)) THEN
    ;
    cmdctr = SCCMDCNT
    /SCATSSTART BUFA
    ;
    WAIT UNTIL (SCCMDCNT .EQ. MOD((cmdctr + 1),256))
    WAIT UNTIL (SCATSCURNUM .EQ. 1) ; Verify buf A
    WAIT UNTIL (SCATPSTATE .EQ. 2) ; Verify execution
    ;
ELSEIF ((ans .EQ. "A") .AND. (SCATPSTATE .EQ. 2)) THEN
    ;
```

```
     ASK "Buffer A is currently executing a load.  Hit 'OK' to kill proc!"
     GOTO STOPPROC
     ;
ELSEIF ((ans .EQ. "B") .AND. (SCATPSTATE .NE. 2)) THEN
     ;
     cmdctr = SCCMDCNT
     /SCATSSTART BUFB
     ;
     WAIT UNTIL (SCCMDCNT .EQ. MOD((cmdctr + 1),256))
     WAIT UNTIL (SCATSCURNUM .EQ. 2) ; Verify buf B
     WAIT UNTIL (SCATPSTATE .EQ. 2) ; Verify Execution
     ;
ELSEIF ((ans .eq."B") .and. (SCATPSTATE .EQ. 2)) THEN
     ;
     ASK "Buffer B is currently executing a load.  Hit 'OK' to kill proc!"
     GOTO STOPPROC
     ;
ELSE
     GOTO OUT
ENDIF
;
GOTO OUT
;
STOPPROC:
;
;
REM; To stop the current ATS from executing and then start the new ATS,
REM; Perform the following steps at the STOL prompt:
REM;      1.  Type /SCATSSTOP
REM;      2.  Type START TCATPSTART
WAIT;
;
OUT:
END:
SHOVAL "PROCEDURE TCATSSTART IS COMPLETE"
ENDPROC
```

```
PROC SCPWRDOWN
;-------------------------------------------------------------------------
;
;   MISSION: GLAST
;
;   TITLE:   GLAST Spacecraft Power Down Procedure
;
;   AUTHOR:  Rick Saylor, AlliedSignal Technical Services Corporation
;
;   DATE:    January 24, 2000
;
;   PURPOSE: This procedure will power down the spacecraft
;
;-------------------------------------------------------------------------
;
;   RESPONSIBLE LEAD:  Tim Trenkle, Triana Lead System Engineer
;
;-------------------------------------------------------------------------
;
;   PROC CALL OUTS:
;
;       NAME:              ACTIVITY:
;     ------------        ----------------------
;     status_ddr          Verifies the DDR recording status
;     totlmon             Turns on telemetry at specified rate
;     egsepwroff          Disables the output on the EGSE power supplies
;     stop_ddr            Stops recording on the DDR
;
;-------------------------------------------------------------------------
;
;   REVISION HISTORY:
;
;    $Id$ $Name$
;
;    $Log$
;
;-------------------------------------------------------------------------
;
;   ASSOCIATED PROCEDURE:
;
;-------------------------------------------------------------------------
;
;   EXECUTION TIME:
;
;*************************************************************************
;
;---------------------- DEFINE PROCEDURE VARIABLES -----------------------
;
SHO "Revision: $Id$ $Name$"
;
LOCAL ans1, ans2, x1, x2
LOCAL log_info          ; Holds log info for power down log
LOCAL pktcnt            ; Packet counter
LOCAL time              ; Time tag for log information
LOCAL cmd_cnt           ; Command counter
LOCAL cmd_err           ; Command error counter
LOCAL pul_cnt           ; Pulse counter
;
;
;---------------------- INITIALIZED VARIABLES ----------------------------
;
ans1 = "X"
ans2 = "X"
x1 = "X"
x2 = "X"
;
;---------------------- CHECK FOR ITOS LOG STATUS ------------------------
;
IF (gbl_logevent .NE. "ON") THEN
    ;
    ;  Start the daily event log
    LOG >>
(CONCAT("EVENT_",SUBSTR(p@GBL_WALLCLOCK,1,2),"_",SUBSTR(p@GBL_WALLCLOCK,4,3),"_",SUBSTR(p@GBL_WALLCLOC
K,8,2),SUBSTR(p@GBL_WALLCLOCK,11,2),".LOG") )
    ;
ENDIF
;
;------------------------ VERIFY DDR STATUS ------------------------------
;
START STATUS_DDR   ; Verify DDR Status
```

```
;
;---------------------- TURN OFF CONFIGMONS ------------------------------
;
CFGMON CLEAR ALL
;
CFGMON1="OFF"
CFGMON2="OFF"
CFGMON3="OFF"
CFGMON4="OFF"
CFGMON5="OFF"
CFGMONUHUB="OFF"
;
;---------------------- ITOS/SCAT CHECKS ---------------------------------
;
ITOSCONFIG:
;
IF (UPPERCASE(GBL_TM_CONNECT) .EQ. "DISCONNECTED") THEN
    ;
    ENABLE TLM  ; Enable ITOS telemetry processor
    WAIT 2
    ;
    AC NAME(TO_TLM_HOST)
    ;
    WAIT UNTIL (UPPERCASE(GBL_TM_CONNECT) .EQ. "CONNECTED"))
    WAIT UNTIL (UPPERCASE(GBL_ACQUIRE) .NE. "OFF"))
    WAIT 2
    ;
    SHO "Telemetry connection made to SCAT"
    ;
ENDIF
;
IF (GBL_CM_CONNECT .EQ. "DISCONNECTED") THEN
    ;
    ENABLE CMD
    WAIT UNTIL (GBL_CM_CONNECT .EQ. "CONNECTED")
    SHO "Command Connection made to SCAT"
    ;
ENDIF
;
/MODE BYPASS OFF
/MODE STEP1
/MODE REXMIT OFF
;
WAIT UNTIL (GBL_BP_MODE .EQ. 0)
WAIT UNTIL (GBL_CMD_MODE .EQ. 3)
WAIT UNTIL (GBL_RETRLIM_0 .EQ. 0)
;
pktcnt = H001CNT
;
WAIT UNTIL (H001CNT .NE. pktcnt)   ; Check for telemetry update
;
;----------------------- S/C TIME UPDATE ---------------------------------
;
TIMEJAM:
;
;ASK "VERIFY THAT THE TIME CODE GENERATOR IS SET TO GMT OR DESIRED SETTING"
;
;DATE SYNC  ; Grab the GMT / Simulated Time
;
;WAIT 2
;
;cmd_cnt = SHTCCMDCNT
;cmd_err = SHTCERRCNT
;
;/TCTIMEJAM GMT=p@GBL_WALLCLOCK ; Jam Spacecraft Time
;
;WAIT UNTIL (SHTCCMDCNT .EQ. (MOD((cmd_cnt + 1), 256)))
;WAIT UNTIL (SHTCERRCNT .EQ. cmd_err)
;
;WAIT UNTIL (HKEVENTCODE .EQ. 50)  ; Verify clock adjust event was generated
;
;
;------------------------ BULK MEMORY RECORDER DUMP -----------------------
;
;  TBD
;  This section will dump the spacecraft recorder, ACS, ENG & Events
;
;------------------------ VERIFY IRWA STATUS ------------------------------
;
```

```
IF ((p@ARW1PWRST .EQ. "OFF") .OR. (p@ARW2PWRST .EQ. "OFF") .OR. (p@ARW3PWRST .EQ. "OFF") .OR.
(p@ARW4PWRST .EQ. "OFF")) THEN
    ;
    /AWHLPWR1 ON
    /AWHLPWR2 ON
    /AWHLPWR3 ON
    /AWHLPWR4 ON
    ;
    WAIT UNTIL (p@ARW1PWRST .EQ. "ON")
    WAIT UNTIL (p@ARW2PWRST .EQ. "ON")
    WAIT UNTIL (p@ARW3PWRST .EQ. "ON")
    WAIT UNTIL (p@ARW4PWRST .EQ. "ON")
    ;
ELSE
    ;
    SHO "WHEELS ARE ALREADY POWERED ON"
    ;
ENDIF
;
SHO "SPIN DOWN ALL WHEELS TO 0 r/s"
;
/AOVSPEED DWELL=65000, RATE1=0, RATE2=0, RATE3=0, RATE4=0
;
WAIT UNTIL ((ACRWSPEED1 .LT. 10.0) .AND. (ACRWSPEED1 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED2 .LT. 10.0) .AND. (ACRWSPEED2 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED3 .LT. 10.0) .AND. (ACRWSPEED3 .GT. -10.0))
WAIT UNTIL ((ACRWSPEED4 .LT. 10.0) .AND. (ACRWSPEED4 .GT. -10.0))
;
/AWHLPWR1 OFF
/AWHLPWR2 OFF
/AWHLPWR3 OFF
/AWHLPWR4 OFF
;
WAIT UNTIL (p@ARW1PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW2PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW3PWRST .EQ. "OFF")
WAIT UNTIL (p@ARW4PWRST .EQ. "OFF")
;
SHO "ALL WHEELS ARE POWERED OFF"
;
;------------------------ ACS SIMULATOR SECTION ---------------------------
;
;  TBD
;
;
;------------------------ RETURN TO HARDLINE COMMANDING -------------------
;
;
IF (cmdmode .EQ. "RF") START RF_CMDOFF
;
IF (tlmmode .EQ. "RF") START RF_TLMOFF
;
;
;------------------------ BATTERY OFF-LINE --------------------------------
;
IF (batmode .EQ. "SIM") THEN
    ;
    START PBATSIMOFF          ; Disable battery simulator
    ;
ELSEIF (batmode .EQ. "BAT") THEN
    ;
    START PBATOFF             ; Disable flight/Qual. battery
    ;
ENDIF
;
;
;------------------------ INSTRUMENT PRE TURN OFF -------------------------
;
;  TBD
;  Will configure instrument for turn off
;
;------------------------ SPACECRAFT POWER OFF ----------------------------
;
WAIT 1
;
/MODE BYPASS ON
;
/TOXMITPWR OFF
;
WAIT UNTIL (p@UHXMITERST .EQ. "OFF")
```

```
;
/PNNISTARPWR DISABLE
WAIT UNTIL (p@PNNISTARPWR .EQ. "OFF")
;
/PNEPICPWR DISABLE
WAIT UNTIL (p@PNEPICPWR .EQ. "OFF")
;
/PNPMHVON DISABLE
WAIT UNTIL (p@PNPMHVON .EQ. "OFF")
;
/PNPMHVOFF ENABLE
WAIT UNTIL (p@PNPMHVOFF .EQ. "OFF")
;
pul_cnt = PNPMHVPULSECNT
/PNPMHVPULSE
;
WAIT UNTIL (PNPMHVPULSECNT .EQ. pul_cnt + 1)
;
/PNPMHVOFF DISABLE
WAIT UNTIL (p@PNPMHVOFF .EQ. "OFF")
;
/PNPMPHAPWR DISABLE
WAIT UNTIL (p@PNPMPHAPWR .EQ. "OFF")
;
/PNHTR1PWR DISABLE
/PNHTR1PWR DISABLE
WAIT UNTIL (p@PNHTR1PWR .EQ. "OFF")
WAIT UNTIL (p@PNHTR1PWR .EQ. "OFF")
;
WAIT 2
;
IF (sasstat .EQ. "ON") THEN
     START SASOFF    ; Disable SAS Power supplies
ELSE
     START EGSEPWROFF   ; DISABLE EGSE POWER SUPPLIES
ENDIF
;
sasstat = "OFF"
;
;
ASK "HIT 'OK' AFTER VERIFYING S/C POWER IS OFF"
;
START STOP_DDR     ; Stop DDR recording
;
;----------------------- POWER NODE LAPTOP POWER DOWN ---------------------
;----------------------- AND STRIP CHART TERMINATION ----------------------
;
OPEN (50) LA-NINA:45000
WRITE (50) "START STRIP_CHARTOFF"
CLOSE (50)
;
ASK "QUIT POWER NODE MONITOR SOFTWARE AND SHUTDOWN LAPTOP"
;
;
;
;------------------------ UART PWR DOWN -----------------------------------
;
PAGE UART_SHUTDOWN    ; Steps for power down the EGSE rack
;
ASK "HIT 'OK' AFTER UART IS POWERED OFF"
;
PAGE CLEAR UART_SHUTDOWN
;------------------------ EGSE PWR DOWN -----------------------------------
;
PAGE EGSE_SHUTDOWN    ; Steps for power down the EGSE rack
;
ASK "HIT 'OK' AFTER EGSE RACK IS POWERED OFF"
;
PAGE CLEAR EGSE_SHUTDOWN
;
;
ASK "Power off all SAS power supplies and turn off SAS rack in back"
;
;------------------------ ARCHIVE STATUS ----------------------------------
;
ARCHIVEST:
;
ARCHIVE STOP ALL     ; Stops all telemetry archiving
;
ARCHIVEST = "OFF"
```

```
;
LOG STOP             ; Stops log event recording
;
LOGSTATUS = "OFF"
;
archive_status = "OFF"
gbl_logevent = "OFF"
;
;------------------------ TIMEON SEQPRT -----------------------------------
;
SEQPRT CLEAR TIMEON    ; Turns off timeon seqprt
;
;------------------------ LOG INFORMATION ---------------------------------
;
START weeklylog        ; Start weeklylog procedure to calculate timeon  data
;
;
time = SUBSTR(p@GBL_WALLCLOCK, 1, 15)
;
log_info = CONCAT(time, ",  SCPWRDOWN")
;
OPEN (5) "/home/triana/SC_LOGS/scpwrdown.log"
WRITE (5) log_info
CLOSE (5)
;
;------------------------ DISABLE ITOS CONFIGURATION ----------------------
;
DISABLE TLM         ; Disables telemtry controller
;
WAIT 1
;
;--------------------------------------------------------------------------
;
END:
SHO "PROCEDURE SCPWRDOWN COMPLETE."
ENDPROC
```